Data Hiding with contrast enhancement by using RDH Algorithm

[1]Ashwini R. Gaykar ,[2]Prof. S.M.Rokade
[1]Student ME Computer Engineering, [2]H.O.D. Dept. of Computer Engineering
[1]Savitribai Phule Pune University, [2]Savitribai Phule Pune University
[1]SVIT, Chincholi, [2] S.V.I.T. Chincholi, Nashik, India

***Abstract*- Now a days more and more attention is paid to reversible data hiding (RDH) for encrypted images, since it gives the benefit that the original cover can be losslessly recovered after embedded data is extracted and as well as protects the image content's confidentiality. Instead of trying to keep the PSNR value high, the proposed algorithm enhances the contrast of a host image to improve its visual quality. the algorithm achieves image contrast enhancement by RDH. The highest two bins in the histogram are selected for data embedding so that histogram equalization can be performed by repeating the process. The side information is embedded along with the message bits into the host image so that the original image is completely recoverable .Reversible data hiding (RDH) algorithm is proposed for digital images.**

***Keywords* - Reversible data hiding, Histogram data modification, Image Encryption,Contrast enhancement**

I. INTRODUCTION

REVERSIBLE DATA HIDING (RDH) has been intensivelystudied in the community of signal processing. Also referred as invertible or lossless data hiding, RDH is to embed a piece of information into a host signal to generate the marked one, from which the original signal can be exactly recovered after extracting the embedded data. The hiding rate and the marked image quality are important metrics while evaluating the performance of a RDH algorithm. To measure the distortion, the peak signal-to-noise ratio (PSNR) value of the marked image is often calculated. we aim at inventing a new RDH algorithm to achieve the property of contrast enhancement instead of just keeping the PSNR value high. Firstly, the two peaks (i.e. the highest two bins) in the histogram are found out. The bins between the peaks are unchanged while the outer bins are shifted outward so that each of the two peaks can be split into two adjacent bins. To increase the embedding capacity, the highest two bins in the modified histogram can be further chosen to be split, and so on until satisfactory contrast enhancement effect is achieved. To avoid the overflows and underflows due to histogram modification, the bounding pixel values are pre-processed and a location map is generated to memorize their locations.

In theoretical aspect, Kalker and Willems [1] established a rate-distortion model for RDH, through which they proved the rate-distortion bounds of RDH for memory less covers and proposed a recursive code construction which, however, does no approach the bound. Zhang *et al* [2] improved the recursive code construction for binary covers and proved that this construction can achieve the rate-distortion bound as long as the compression algorithm reaches entropy, which establishes the equivalence between data compression and RDH for binary covers.

In practical aspect, many RDH techniques have emerged inrecent years. A more popular method is based on difference expansion (DE) [3] in which the difference of each pixel group is expanded, e.g., multiplied by 2, and thus the least significant bits (LSBs) of the difference are all-zero and can be used for embedding messages. Another promising strategy for RDH is histogram shift (HS) [4], in which space is saved for data embedding by shifting the bins of histogram of gray values. With regard to providing confidentiality for images, encryption is an effective and popular means as it converts the original and meaningful content to incomprehensible one. Although few RDH techniques in encrypted images have been published yet, there are some promising applications if RDH can be applied to encrypted images. In [9], Hwang *et al.* advocated a reputation-based trust-management scheme enhanced with data coloring (a way of embedding data into covers) and software watermarking, in which data encryption and coloring offer possibilities for upholding the content owner's privacy and data integrity. Obviously, the cloud service provider has no right to introduce permanent distortion during data coloring into encrypted data. Thus, a reversible data coloring technique based on encrypted data is preferred. Suppose a medical image databaseis stored in a data center, and a server in the data center can embed notations into an encrypted version of a medical image through a RDH technique. With the notations, the server can manage the image or verify its integrity without having the knowledge of the original content, and thus the patient's privacy is protected. On the other hand, a doctor, having the cryptographic key, can decrypt and restore the image in a reversible manner for the purpose of further diagnosing. Hong *et al.* [11] reduced the error rate of Zhang'smethod by fully exploiting the pixels in calculating the smoothness of each block and using side match. The extraction and recovery of blocks are performed according to the descending order of the absolute smoothness difference between two candidate blocks and recovered blocks can further be used to evaluate the smoothness of unrecovered blocks, which is referred to as side match

II. RELATED WORK

The methods proposed previously can be summarized as the framework, "vacating room after encryption (VRAE)", as illustrated in Fig. 1(a).

In this framework, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider (e.g., a database manager) and the data hider an embed some auxiliary data into the encrypted image by losslessly vacating some room according to a data hiding key. Then a receiver, maybe the content owner himself or an authorized third party can extract the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key.
As shown in Fig. 1(b), the content owner first reserves enou-
ghspace on original image and then converts the image into its encrypted version with the encryption key. Now, the data embeddin g process in encrypted images is inherently reversible for thedata hider only needs to accommodate data into the spare spacepreviou s emptied out. The data extraction and image recoveryare identical to that of Framework VRAE. Obviously, standardRDH algori thms are the ideal operator for reserving room before encryption and can be easily applied to Framework RRBEto achieve better performance compared with techniques fromFramework VRAE. This is because in this new framework, wefollow the customary i dea that first losslessly compresses the redundant image content (e.g., using excellent RDH techniques)and then encrypts it with re spect to protecting privacy.

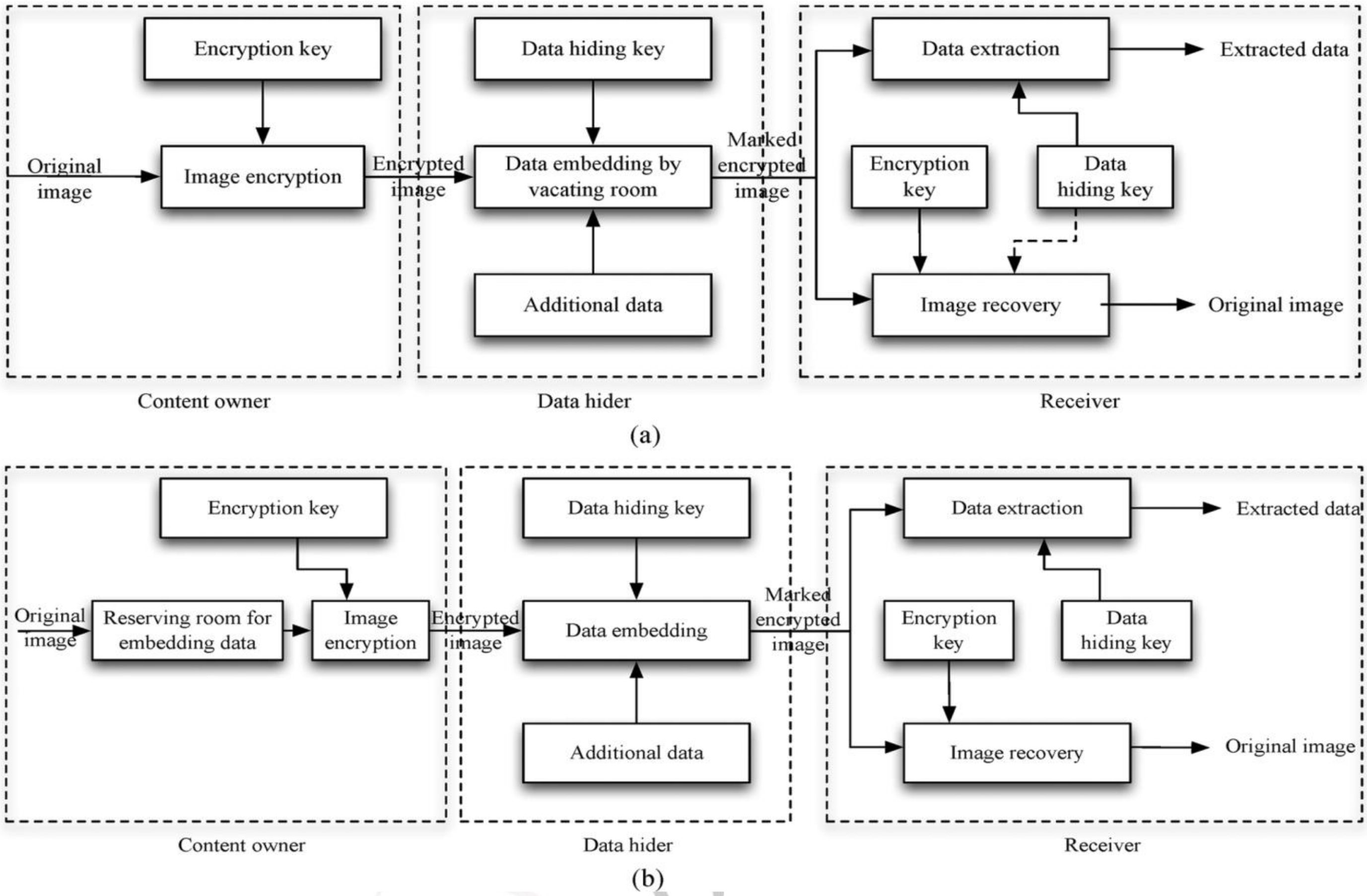

Fig. 1. Framework: "vacating room after encryption (VRAE)" versus framework: "reserving room before encryption (RRBE)." (Dashed line in (a) states that the need of data hiding key in image recovery varies in different practical methods). (a) Framework VRAE. (b) Framework RRBE.
we elaborate a practical method based on the Framework "RRBE", which primarily consists of four stages: genertion of encrypt ed image, data hiding in encrypted image, dataextraction and image recovery.

A. Generation of Encrypted Image
Actually, to construct the encrypted image, the first stage canbe divided into three steps: image partition, self reversible embeddingfollowed by image encryption. At the beginning, imagepartition step divides original image into two parts A and B.

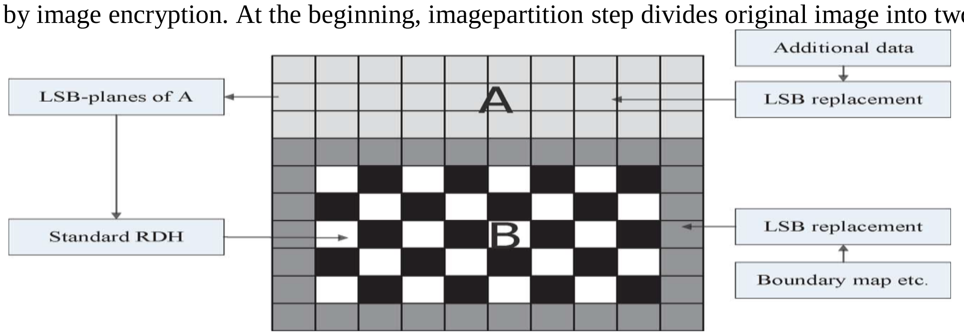

Fig. 2.Illustration of image partition and embedding process.

1) *Image Partition:* The operator here for reserving room before encryption is a standard RDH technique, so the goal ofimage partition is to construct a smoother area on B , on whichstandard RDH algorithms can achieve betterperformance
2) *Self-Reversible Embedding:* The goal of self-reversibleembedding is to embed the LSB-planes of A and B. By bidirectional histogram shift, some messages can be embeddedon each error sequence. That is, first divide the histogramof estimating errors into two parts, i.e., the left part andthe right part, and search for the highest point in each part,
3) *Image Encryption:* After rearranged self-embeddedimage, denoted by , is generated, we can encrypts to constructthe encrypted image, denoted by .

B. Data Hiding in Encrypted Image

Once the data hider acquires the encrypted image , he canembed some data into it, although he does not get access to the original image.

C. Data Extraction and Image Recovery

Since data extraction is completely independent from imagedecryption, the order of them implies two different practical applicatio ns.

1) Case 1: Extracting Data From Encrypted Images: Tomanage and update personal information of images which areenc rypted for protecting clients' privacy, an inferior databasemanager may only get access to the data hiding key and have tomanipu late data in encrypted domain. The order of data extraction before image decryption guarantees the feasibility of ourwork in this case.

When the database manager gets the data hiding key, he candecrypt the LSB-planes and extract the additional data by directly reading the decrypted version. When requesting forupdating information of encrypted images, the database manager, then, updates information through LSB replacement andencrypts updated information according to the data hiding keyall over a gain. As the whole process is entirely operated on encrypted domain, it avoids the leakage of original content.

*2) Case 2: Extracting Data From Decrypted Images:*In Case1, both embedding and extraction of the data are manipulated inencr ypted domain. On the other hand, there is a different situation that the user wants to decrypt the image first and extracts thedata from the decrypted image when it is needed. The followingexample is an application for such scenario. Assume Alice outsourced her i mages to a cloud server, and the images are encrypted to protect their contents. Into the encrypted images, thecloud server mar ks the images by embedding some notation,including the identity of the images' owner, the identity of thecloud server and time stamps, to manage the encrypted images.The order of image decryption before/without data extraction is perfectly suitable for this case.

III. THE PROPOSED SCHEMES

The procedure of the proposed algorithm is illustrated inFig. 3. Given that totally pairs of histogram bins are to besplit for data embedding, the **embedding** procedure includesthe following steps:

1. Pre-process: A locationmap is generated to record the locations of those pixelsand compressed by the JBIG2 standard [11] to reduce itslength.
2. The image histogram is calculated without counting thefirst 16 pixels in the bottom row.
3. Embedding: The two peaks (i.e. the highest two bins) in thehistogram are split for data embedding is applied to every pixel counted in the histogram. Then the two peaksin the *modified* histogram are chosen to be split, and so onuntil pairs are split. The bitstream of the compressed locationmap is embedded before the message bits (binaryvalues). The value of , the length of the compressed locationmap, the LSBs collected from the 16 excluded pixels and the previous peak values are embedded with the lasttwo peaks to be split.
4. The lastly split peak values are used to replace the LSBs ofthe 16 excluded pixels to form the marked image.

The **extraction** and **recovery** process include the followingsteps:

1. The LSBs of the 16 excluded pixels are retrieved so that the values of the last two split peaks are known.
2. The data embedded with the last two split peaks are extracted so that the value of , the lengthof the compressed location map, the original LSBs of 16excluded pixels, and the previously split peak values areknown. Then the recovery operations are carried out byprocessing all pixels except the 16 excluded . The process of extraction and recovery is repeateduntil all of the split peaks are restored and the data embeddedwith them are extracted.
3. The compressed location map is obtained from the extractedbinary values and decompressed to the original size.
4. With the decompressed map, those pixels modified in preprocessare identified. Among them, a pixel value is subtractedby if it is less than 128, or increased by otherwise.

To comply with this rule, the maximum value ofis 64 to avoid ambiguity. At last, the original image is recoveredby writing back the original LSBs of 16 excludedpixels.

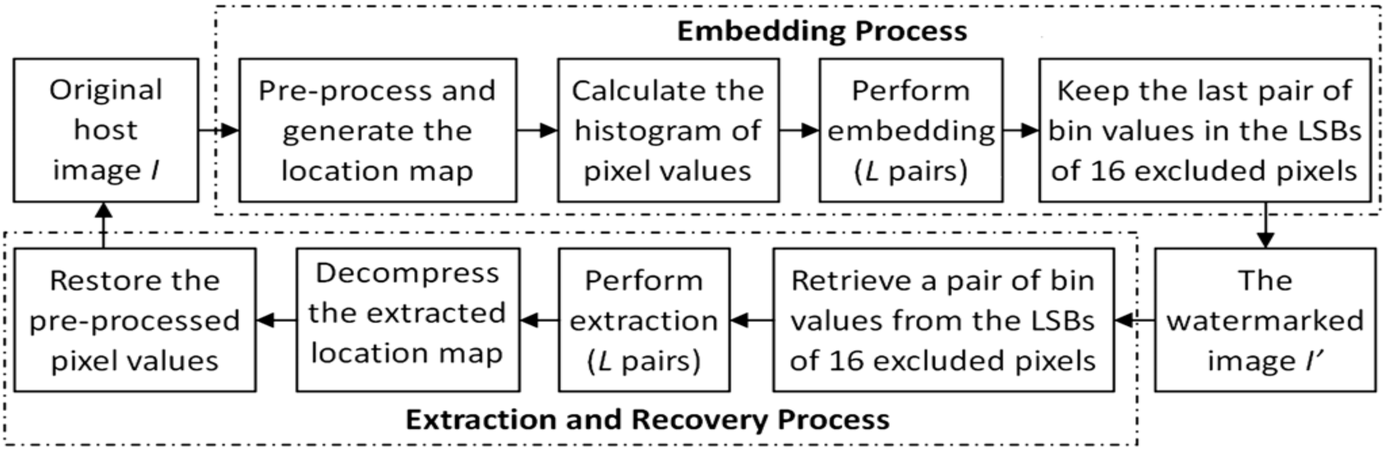

Fig. 3.Procedure of the proposed RDH algorithm.

IV. RESULT ANALYSIS

The original and marked images of "Lena" is shown in Fig.The marked imageswere obtained by splitting 10, 15 and 20 pairs of histogrampeaks for data embedding, respectively. It can be seen that theembedded data were invisible in the contrast-enhanced images.The more histogram peaks were split for data embedding, themore contrast enhancement effect was obtained. Although thePSNR value of the contrast-enhanced images decrease with thedata hiding rate, the visual quality has been preserved, as shownin Fig. 4.

Besides the PSNR value, the relative contrast error (RCE),relative entropy error (REE), relative mean brightness error(RMBE) and relative structural similarity (RSS) used were calculated between the original and contrast-enhancedimages to evaluate the enhancement effect and image quality.The RCE and REE values greater than 0.5 indicate the enhancedcontrast and increased image data, respectively. The less differencein mean brightness from the original image, the closerRMBE is to 1. The greater the structural similarity betweenthem, the closer RSS is to 1. We further compare the proposedalgorithm with three MATLAB functions used for image contrastenhancement, i.e. *imadjust*, *histeq*, and *adapthisteq*. TheMATLAB routines were applied on each test image with thedefault settings. For each of the contrast-enhanced images, thefive evaluation values were calculated, including RCE, REE,RMBE, RSS and PSNR.

Fig. 4.The original and contrast-enhanced images of "Lena" by splitting 10, 15and 20 pairs of histogram peaks in the proposed algorithm. (a) Original imageof "Lena". (b) 10 pairs: 0.185 bpp, 29.10 dB. (c) 15 pairs: 0.268 bpp, 25.97 dB.(d) 20 pairs: 0.345 bpp, 24.91 dB.

V. CONCLUSION

In this letter, a new reversible data hiding algorithm has beenproposed with the property of contrast enhancement.Previous methods implement RDH in encrypted images by vacating room afterandbefore encryption,. Thus the data hider can benefit from the extra spaceemptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditionalRDH techniques for plain images and achieve excellent performance without loss of perfect secre cy. Improving the algorithm robustness,and applying it to the medical and satellite images for the bettervisibility, will be our future work.

VI. ACKNOWLDGEMENT

I take this opportunity to express my hearty thanks to all those who helped me in the completion of the Paper. I express my deep sense of gratitude to my guide Prof. S.M.Rokade, HOD and Asst.Prof., Computer Engineering Department, Sir Visvesvaraya Institute of Technology, Chincholi for his guidance and continuous motivation. I great fully acknowledge the help provided by him on many occasions, for improvement of this paper with great interest. I would also like to thank to Prof. M.M.Naoghare P.G. Co-ordinator for her great support and excellent guidance.

VII. REFERENCES

[1] T. Kalker and F.M.Willems, "Capacity bounds and code constructionsfor reversible data-hiding," in *Proc. 14th Int. Conf. Digital Signal Processing(DSP2002)*, 2002, pp. 71–76.

[2] W. Zhang, B. Chen, and N. Yu, "Improving various reversible datahiding schemes via optimal codes for binary covers," *IEEE Trans.Image Process.*, vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[3] J. Tian, "Reversible data embedding using a difference expansion,"*IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896,Aug. 2003.

[4] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," *IEEETrans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar.2006.

[5] D.M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3,pp. 721–730, Mar. 2007.

[6] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarkingbased on adaptive prediction-error expansion and pixel selection,"*IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec.2011.

[7] L. Luo*et al.*, "Reversible imagewatermarking using interpolation technique,"*IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193,Mar. 2010.

[8] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversiblewatermarking algorithm using sorting and prediction," *IEEE Trans.Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.

[9] K. Hwang and D. Li, "Trusted cloud computing with secure resourcesand data coloring," *IEEE Internet Comput.*, vol. 14, no. 5, pp. 14–22,Sep./Oct. 2010.

[10] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K.Ramchandran, "On compressing encrypted data," *IEEE Trans. SignalProcess.*, vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[11] W. Hong, T. Chen, and H.Wu, "An improved reversible data hiding inencrypted images using side match," *IEEE Signal Process. Lett.*, vol.19, no. 4, pp. 199–202, Apr. 2012.

[12] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEETrans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.